Pacific Graphics (2011)
Bing-Yu Chen, Jan Kautz, Tong-Yee Lee, and Ming C. Lin (Editors)

Short Papers

Sketch-based breaking waves

Guijuan Zhang [1], Gaojin Wen[1], Dengming Zhu[2], Dianjie Lu [1] and Shengzhong Feng [1]

[1] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[2] Institute of Computing Technology, Chinese Academy of Sciences

Abstract

We present a sketching method for animating breaking waves in this paper. Our approach allows an animator to control physically-based wave breaking effects by drawing the outline of the wave shape intuitively. To do this, we provide a feature-based keyframe design model. We can sketch the wave shape roughly and describe the wave shape as a unified volume representation. Moreover, we propose an efficient keyframe matching model to transport current fluid state to the target state with minimum energy. The solution to the keyframe matching model is used to compute the external force field for the Navier-Stokes equations. Our method is straightforward to implement and convenient to produce interesting wave breaking behaviors. The experimental results show that our method is easy and intuitive to use.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Breaking waves are most common natural phenomena in our daily life. They appear frequently in many disaster films, such as “*2012*”, “*The day after tomorrow*” and so on . In such applications, special effects about wave breaking often require both high degree of realism and freedom of liquid control. Although the physically-based fluid animation model can provide nice results, it suffers from the high nonlinearity of the motion equation. Therefore, generating controllable wave breaking effects is still one of the most challenging problems in the field of physically-based fluid animation.

It is expected that animators could describe desirable wave behavior freely by sketching out the outline of breaking waves. Then, the fluid will move in accordance with the sketched outlines. However, it is difficult to do this for two reasons. First, the wave breaking results are overly dependent on the keyframes. The keyframe designing involves a very tedious process since the amorphous phenomena tend to undergoes intangible motion. Second, the dynamics of breaking waves should be considered when computing the fluid velocity to capture full 3D wave behaviors. Therefore, developing a method that overcomes the above difficulties and provides an easy-to-use interface is of great importance for wave animation.

We propose a sketching method for animating breaking waves in this paper. The method allows users to control wave breaking intuitively via outlining wave shapes freely in keyframes. Therefore, our method could reduce the labor of animators significantly. The framework of our method includes two novel parts (see Figure 1): the feature-based model for keyframe designing and the energy minimizing model for keyframe matching. We will give the details in the following sections.

2. Related Work

Early work in the field of ocean simulation tends to adopt the knowledge of oceanographic phenomenology and model the ocean surface using parametric function [FR86, GS00, MWM87, HNC02, JBS03]. A comprehensive survey can be found in [DCGG11]. Such parametric representation is unable to describe the important features of breaking waves (i.e., overturning of the wave surface). Thus, animating wave breaking effects is virtually impossible for the semi-empirical methods.

Another group of methods use 2D shallow water equations to model breaking waves [TMFSG07]. In contrast to the parametric methods, the shallow water equations can obtain a velocity field that improves realism of wave motion.

In order to simulate breaking waves with full 3D fluid be-

DOI: 10.2312/PE/PG/PG2011short/061-066

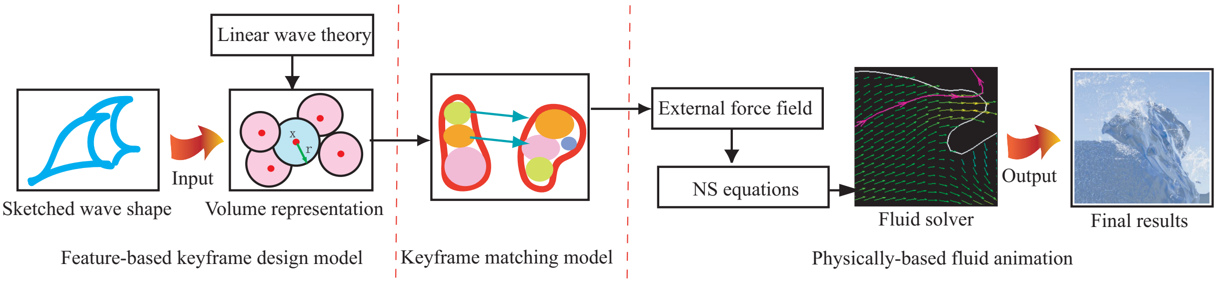

Figure 1: The framework of our method.

haviors, models of physically-based fluid animation should be considered. Examples include particle level set method [EMF02] and its improved techniques [KCC*06], two-way coupled SPH (Smoothed Particle Hydrodynamics) and PLS (Particle Level Set) method [LTKF08], surface tension model [HK05, TWGT10], explicit method [WTGT10, BBB10] and so on.

Building on the physically-based fluid models, the forward control methods [FM97, FF01, REN*04] as well as the inverse control methods [MTPS04, SY05, TKPR06, ZZQW11] are introduced to control the liquid motion. The forward control methods require a “try-and-error” process. Although the inverse methods solve the above problem, they are difficult to produce wave breaking effects since the keyframes designing is rather tedious in wave animation.

Recently, Darles et al. [DCG10] propose a wave breaking model based on particles. To animate breaking waves under grid-based Eulerian approach, Mihalef et al. [MMS04] present a slice method. The method constructs the initial wave shape and velocity field based on a library of breaking waves. Therefore, animators are enabled to animate breaking waves by controlling the shape and the timing of breaking. Losasso et al. [LTKF08] extend the slice method to a two-way coupled SPH and PLS model.

3. Feature-based Keyframe Design Model

3.1. Wave Modeling Operators

The wave modeling operators are entirely inspired by the concepts in solid modeling [RR99]. Formally, set $S = sweep(L_1, L_p)$, where L_1 (a close spline curve) is the 2D silhouette of wave shape and L_p is the path for sweeping. The wave shape S is the result of the sweeping operation *sweep*. The *sweep* operator here converts the input into 3D wave shapes automatically. Figure 2 gives an example.

3.2. The Unified Volume Representation

The rough sketching task only provides approximate wave shapes. To counteract the inaccuracy, we introduce a unified volume representation here. Let $\mathcal{K} = \{k_1, k_2, \cdots, k_l\}$ be l keyframes used in the animation. $k_i = (S_i, \boldsymbol{u}_i)$ denotes the ith keyframe. In k_i, S_i represents the wave shape and $\boldsymbol{u}_i$ denotes the velocity field. Let $C_t = \{c_t^1, c_t^2, \cdots, c_t^n\}$ be the volume representation of k_i. Each element c_t^i in C_t is a cluster that contains several fluid cells. The cluster can be represented as a sphere with four attributes: position $\boldsymbol{x}(c_i)$, control radius $r(c_i)$, velocity $\boldsymbol{u}(c_i)$ and volume $v(c_i)$. These attributes contain basic shape and velocity information defined in the keyframes. Obviously, the more clusters are used, the more accurate representation can be achieved.

In order to describe the keyframes into the unified volume representation, we approximate wave shapes with hierarchical spheres according to Bradshow’s sphere tree method [BO02]. Each sphere has attributes of position $\boldsymbol{x}$, radius r, volume v and velocity $\boldsymbol{u}$. We compute the velocity for the sphere according to the linear wave theory [MMS04].

4. Energy-minimizing Model for Keyframe Matching

In order to transform the current state to the target, we need a way to determine the fluid velocity field $\boldsymbol{u}$. We present an energy-minimizing model to find the matching solution among the source clusters and the target clusters. The solution will be used to compute the velocity field $\boldsymbol{u}$ that satisfies the minimum energy condition.

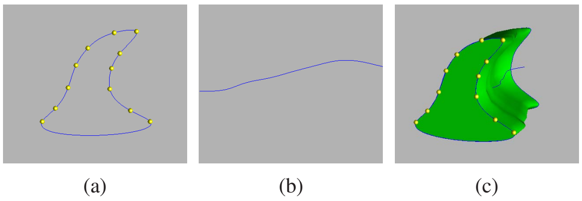

Figure 2: Shape designing by the *sweep* operation. (a) The 2D silhouette of a wave shape; (b) The path for sweeping; (c) The final wave shape.

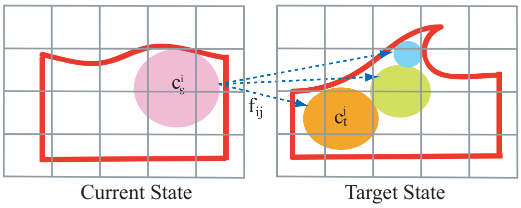

Figure 3: Keyframe matching in our method. We use circles to denote clusters. The optimal solution of the keyframe matching model is a flow matrix F. It records the transported volumes among the clusters in the current state and the target state. Specifically, the element f_{ij} in F denotes the transported volume between a source cluster c_s^i and a target cluster c_t^j.

4.1. Keyframe matching model

We formulate the keyframe matching problem as an energy-minimizing model. The objective function is defined as the energy consumption of the whole deformation process. We consider two forms of energy here: the kinetic energy and the potential energy. The kinetic energy is measured by the work done in converting current velocity field to that of the target. The potential energy reflects the work done in transporting current shape to its target.

Formally, let $C_s = \{c_s^1, \cdots, c_s^m\}$ be the current state and $C_t = \{c_t^1, c_t^2, \cdots, c_t^n\}$ be the target state. $D = [d_{ij}]$ is the Euclidean distance matrix among the clusters in C_s and C_t. We solve for a flow matrix $F = [f_{ij}]$ that minimize the total energy. As shown in Figure 3, the value of f_{ij} equals to the volume that transported between c_s^i and c_t^j. Thus, the energy consumption of a cluster c_s^i takes the form

$$E(c_s^i) = \sum_{j=1}^{n} f_{ij} s_{ij}, \tag{1}$$

where

$$s_{ij} = d_{ij} + \frac{1}{2}(|\boldsymbol{u}(c_s^i)|^2 - |\boldsymbol{u}(c_t^j)|^2).$$

Note that d_{ij} is the Euclidean distance between the positions of c_s^i and c_t^j, $\boldsymbol{u}$ is the velocity. The first term on the right-hand side denotes the potential energy and the second term is the kinetic energy.

Performing the summation of $E(\cdot)$ over all source clusters yields the objective function of our keyframe matching model

$$\arg\min_F \ Work(C_s, F, C_t) = \sum_i E(c_s^i).$$

Equation (2) shows the definition of $Work(C_s, F, C_t)$ with a set of constraints.

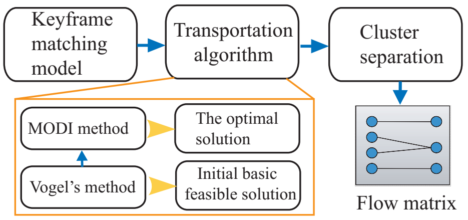

Figure 4: The basic procedure for solving the keyframe matching model. We obtain the optimal solution according to the *transportation algorithm*. The cluster separation step is involved to make sure that each cluster has only one destination.

$$\begin{aligned}
&Work(C_s, F, C_t) = \sum_{i=1}^{m} \sum_{j=1}^{n} s_{ij} f_{ij} \\
&s.t. \\
&f_{ij} \geq 0 \qquad 1 \leq i \leq m, \ \ 1 \leq j \leq n \\
&\sum_{j=1}^{n} f_{ij} < v(c_s^i) \qquad 1 \leq i \leq m \\
&\sum_{i=1}^{m} f_{ij} < v(c_t^j) \qquad 1 \leq j \leq n \\
&\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = min(\sum_{i=1}^{m} v(c_s^i), \sum_{j=1}^{n} v(c_t^j)).
\end{aligned} \tag{2}$$

The first constraint means that the volume transported between two clusters c_s^i and c_t^j is no less than 0. The next two constraints mean that the shipping volume is no more than the cluster's volume itself. The fourth constraint ensures that the total shipping volume from the source clusters to the target clusters is no more than the total volume of both states.

4.2. Keyframe Matching Solution

Actually, the keyframe matching model (2) can be regarded as a well-known *transportation problem* [HL05]. The optimal solution of (2) determines the minimum amount of energy required in the transportation process. Since a source cluster may transport fluid volume to any target cluster, the traditional *transportation problem* returns a flow matrix that records a many-to-many relationship among the source clusters and the target clusters. See an example in Figure 3. We wish to get a one-to-one or many-to-one relationship in the flow matrix because it will be used to derive the force direction for fluid motion. As a result, one source cluster has no more than one destination. To achieve this, we present a tailored method for solving the keyframe matching model as shown in Figure 4.

In our method, the keyframe matching model can be solved in two steps. The first step solves (2) according to the *transportation algorithm* while the second step performs cluster separation over the source clusters. Details about the *transportation algorithm* can be found in [HL05]. In the separation step, we get the modified flow matrix that each source cluster has no more than one target cluster. According to the target number of each source cluster, we separate the original source cluster into several parts and the volume of each newly generated cluster satisfies its shipping demand.

5. Controlled Fluid Motion

We use the modified flow matrix F to construct the external force field. The external force is added to the Navier-Stokes equations so as to yield the final fluid velocity field for wave motion.

5.1. The External Force Field

To transport the current fluid shape to its target while compensating for the velocity difference, we consider the shape feedback force and the velocity feedback force here. The shape feedback force acting on a cluster center $\boldsymbol{x}(c_s^i)$ is

$$\boldsymbol{f}_a(\boldsymbol{x}(c_s^i)) = w_a(\boldsymbol{x}(c_t^j) - \boldsymbol{x}(c_s^i)),$$

where w_a is weight of the shape feedback force, c_t^j is the matching cluster of c_s^i. The velocity feedback force acting on $\boldsymbol{x}(c_s^i)$ is defined as

$$\boldsymbol{f}_v(\boldsymbol{x}(c_s^i)) = w_v(\boldsymbol{u}(c_t^j) - \boldsymbol{u}(c_s^i)),$$

where w_v is the weight of the velocity feedback force, and $\boldsymbol{u}$ is the velocity. Thus, the external force acting on $\boldsymbol{x}(c_s^i)$ is

$$\boldsymbol{f}(\boldsymbol{x}(c_s^i)) = \boldsymbol{f}_a(\boldsymbol{x}(c_s^i)) + \boldsymbol{f}_v(\boldsymbol{x}(c_s^i)). \qquad (3)$$

Next, we compute the external force exerted on a water cell x_c

$$\boldsymbol{f}(x_c) = \sum_i W(|\boldsymbol{x}(c_s^i) - \boldsymbol{x}(x_c)|, \alpha r(c_s^i))\boldsymbol{f}(\boldsymbol{x}(c_s^i)), \qquad (4)$$

where α adjusts the control radius of the kernel function. We choose the values between 1.0 and 2.0 for α in our examples. W is the normalized spline kernel [TKPR06] with support h

$$W(d,h) = \begin{cases} \frac{315}{64\pi h^9}(h^2 - d^2)^3 & \text{if } d < h \\ 0 & \text{if } d \geq h. \end{cases}$$

Equation (4) defines the external force field for wave controlling.

5.2. Algorithm for Breaking Waves

We use the two-way coupled physically-based fluid animation method [LTKF08] to compute wave motion in our approach. In the two-way coupled fluid animation method, the water surface is defined as the zero iso-counter of a signed distance function and it moves under the level set equation

$$\frac{\partial\phi}{\partial t} + \boldsymbol{u}\cdot\nabla\phi = 0,$$

where ϕ is the signed distance function with $|\nabla\phi| = 1$, $\boldsymbol{u}$ is the fluid velocity. Also, the fluid details are represented as particles and modeled by the SPH method. The particles are advected in the fluid velocity field

$$\frac{d\boldsymbol{x}_p}{dt} = \boldsymbol{u}_p,$$

where $\boldsymbol{u}_p$ is the fluid velocity interpolated at the particle location $\boldsymbol{x}_p$. In the two-way coupled fluid animation method, the velocity field at dense fluid regions can be obtained by solving the following Navier-Stokes equations

$$\nabla\cdot\boldsymbol{u} = 0,$$

$$\boldsymbol{u}_t = -\boldsymbol{u}\cdot(\nabla\boldsymbol{u}) + \nu\nabla\cdot\nabla\boldsymbol{u} - \frac{1}{\rho}\nabla p + \boldsymbol{f}, \qquad (5)$$

where ν is the kinetic viscosity, ρ is the density, p is the pressure, and $\boldsymbol{f}$ is the external force field. Moreover, the velocity at diffuse fluid volumes can be computed by solving the SPH model. Note that the pressure term is obtained by rasterizing the particle values onto the background MAC grid and solv-

Algorithm 1: Algorithm for animating breaking waves

```
Input: The initial condition, $\mathcal{K}$, and control parameters.
Output: A sequence of controlled breaking waves.
begin
    $match \longleftarrow true$;
    while animating do
        if match then
            $C_t \longleftarrow FetchKeyframe(\mathcal{K})$;
            if $C_t$ is NULL then
                $\boldsymbol{u} \longleftarrow FluidSover()$;
                continue;
            end
            $match \longleftarrow false$;
            $C_s \longleftarrow VolRep(CurrentState)$;
            $F \longleftarrow SolveMP(C_s, C_t)$;
        end
        $\boldsymbol{f} \longleftarrow GenExternalForce(F)$;
        $\boldsymbol{u} \longleftarrow FluidSolver(\boldsymbol{f})$;
        $\boldsymbol{x}(C_s) \longleftarrow UpdateClusterPos()$;
        $avgdis \longleftarrow AvgDistance(\boldsymbol{x}(C_s), \boldsymbol{x}(C_t))$;
        if $avgdis < d$ then
            $match \longleftarrow true$;
        end
    end
end
```

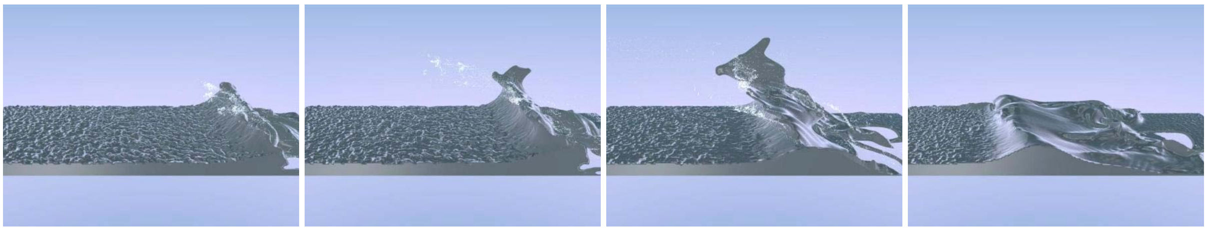

Figure 5: A sequence of a controlled breaking wave.

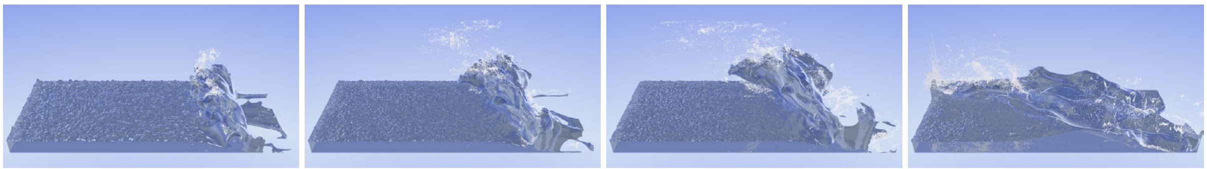

Figure 6: Another sequence of a controlled breaking wave.

ing the poisson equation coupled with (5). Further details are discussed in the literature [LTKF08, EMF02].

Algorithm 1 gives our wave animation method. In this algorithm, $FluidSover(\cdot)$ solves fluid velocity field using the two-way coupled SPH and PLS model, $VolRep(\cdot)$ returns the unified volume representation, $SolveMP(\cdot)$ solves the energy-minimizing keyframe matching model and $GenExternalForce(\cdot)$ computes the external force field. After obtaining the final fluid velocity field, we update the positions of the source clusters in $UpdateClusterPos(\cdot)$ with the Newton's second law. We compute the average distance from current cluster centers to the target ones. The current state is matched to the target when the average distance is below a threshhold.

6. Results

We will give some examples of breaking waves in this section. All the results are gathered on a PC with 2.6 GHz Intel Core 2 Duo CPU and 2GB memory. And the 3D water is rendered using Povray library which is available from http://www.povray.org/. In the following examples, we use $w_a = 0.02$, $w_v = 0.1$ to compute the external force field.

Figure 5 shows an example of a breaking wave. We sketched the wave shape in our keyframe. The four images show the wave motion at different time steps. In this example, the grid size is set to $240 \times 64 \times 96$. As for the initial condition, we use a randomly generated height field to represent the free surface. The wave control model in this paper makes the surface grow away from their original locations and finally forms the wave breaking effects.

Figure 6 shows the motion of another breaking wave. In this example, the path for sweeping operation is a curve. Since the feature-based keyframe design model supports rough sketching, users could design the desired wave shape freely. The grid size in this example also equals to $240 \times 80 \times 96$. In the above examples, we generate wave breaking results via abandoning the external control forces directly in the Navier-Stokes equations. The results show that our approach is easy and intuitive to use while producing desired wave motion.

7. Conclusion and Future Work

We have presented a sketching method for breaking wave animation in this paper. Our method allows users to control wave breaking freely by sketching out the outline of the wave shapes. The rough sketching method benefits from our keyframe design model and keyframe matching model significantly. The keyframe design model facilities the keyframe designing efforts while the keyframe matching model relaxes the dependence of wave animation on the provided keyframes.

Since the control parameters about the external force field are related to motion data, it is difficult for us to design a group of optimal parameters for all kinds of breaking waves. Thus, animators are required to adjust the parameters for a given set of examples. We would like to improve this situation in our future work.

Acknowledgements

This work is supported by the National High-Tech Research and Development Plan of China (863 program) under Grant No.2009AA01A129-2, the Science and Technology Project of Guangdong Province under the grant No.2010A090100028, National Natural Science Foundation of P. R. China under the Grant No.60903116, and Knowledge Innovation Project of the Chinese Academy of Sciences No.KGCX2-YW-131, the Science and Technology

Project of Shenzhen under the grant No. JC200903170443A, ZD201006100023A.

References

[BBB10] BROCHU T., BATTY C., BRIDSON R.: Matching fluid simulation elements to surface geometry and topology. In *SIGGRAPH '10: ACM SIGGRAPH 2010 papers* (New York, NY, USA, 2010), ACM, pp. 1–9.

[BO02] BRADSHAW G., O'SULLIVAN C.: Sphere-tree construction using dynamic medial axis approximation. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2002), ACM, pp. 33–40.

[Dan63] DANTZIG G. B.: Linear programming and extensions.

[DCG10] DARLES E., CRESPIN B., GHAZANFARPOUR D.: A particle-based method for large-scale breaking waves simulation. In *ICCVG (1)* (2010), pp. 316–323.

[DCGG11] DARLES E., CRESPIN B., GHAZANFARPOUR D., GONZATO J.-C.: A survey of ocean simulation and rendering techniques in computer graphics. *Comput. Graph. Forum 30*, 1 (2011), 43–60.

[EMF02] ENRIGHT D., MARSCHNER S., FEDKIW R.: Animation and rendering of complex water surfaces. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2002), ACM, pp. 736–744.

[FF01] FOSTER N., FEDKIW R.: Practical animation of liquids. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM, pp. 23–30.

[FM97] FOSTER N., METAXAS D.: Controlling fluid animation. In *CGI '97: Proceedings of the 1997 Conference on Computer Graphics International* (Washington, DC, USA, 1997), IEEE Computer Society, p. 178.

[FR86] FOURNIER A., REEVES W. T.: A simple model of ocean waves. *SIGGRAPH Comput. Graph. 20* (August 1986), 75–84.

[GS00] GONZATO J.-C., SAËC B. L.: On modelling and rendering ocean scenes. *Journal of Visualization and Computer Animation 11*, 1 (2000), 27–37.

[HK05] HONG J., KIM C.: Discontinuous fluids. *ACM Transactions on Graphics 24* (2005), 915–920.

[HL05] HILLIER F. S. F. S., LIEBERMAN G. J.: *Introduction to operations research / Frederick S. Hillier, Gerald J. Lieberman*, 8th ed ed. New York ; London : McGraw-Hill, 2005. "International Edition"–Cover.

[HNC02] HINSINGER D., NEYRET F., CANI M.-P.: Interactive animation of ocean waves, july 2002.

[JBS03] JESCHKE S., BIRKHOLZ H., SCHUMANN H.: A procedural model for interactive animation of breaking ocean waves. In *WSCG* (2003).

[KCC*06] KIM J., CHA D., CHANG B., KOO B., IHM I.: Practical animation of turbulent splashing water. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2006), Eurographics Association, pp. 335–344.

[LTKF08] LOSASSO F., TALTON J., KWATRA N., FEDKIW R.: Two-way coupled sph and particle level set fluid simulation. *IEEE Transactions on Visualization and Computer Graphics 14*, 4 (2008), 797–804.

[MMS04] MIHALEF V., METAXAS D., SUSSMAN M.: Animation and control of breaking waves. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2004), Eurographics Association, pp. 315–324.

[MTPS04] MCNAMARA A., TREUILLE A., POPOVIĆ Z., STAM J.: Fluid control using the adjoint method. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), ACM, pp. 449–456.

[MWM87] MASTIN G. A., WATTERBERG P. A., MAREDA J. F.: Fourier synthesis of ocean scenes. *IEEE Comput. Graph. Appl. 7* (March 1987), 16–23.

[REN*04] RASMUSSEN N., ENRIGHT D., NGUYEN D., MARINO S., SUMNER N., GEIGER W., HOON S., FEDKIW R.: Directable photorealistic liquids. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2004), Eurographics Association, pp. 193–202.

[RR99] ROSSIGNAC J. R., REQUICHA A. A. G.: Solid modeling, 1999.

[SY05] SHI L., YU Y.: Taming liquids for rapidly changing targets. In *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2005), ACM, pp. 229–236.

[TKPR06] THÜREY N., KEISER R., PAULY M., RÜDE U.: Detail-preserving fluid control. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2006), Eurographics Association, pp. 7–12.

[TMFSG07] THÜREY N., MÜLLER-FISCHER M., SCHIRM S., GROSS M.: Real-time Breaking Waves for Shallow Water Simulations. *Proceedings of the Pacific Conference on Computer Graphics and Applications 2007* (October 2007), 8.

[TWGT10] THÜREY N., WOJTAN C., GROSS M., TURK G.: A multiscale approach to mesh-based surface tension flows. In *SIGGRAPH '10: ACM SIGGRAPH 2010 papers* (New York, NY, USA, 2010), ACM, pp. 1–10.

[WTGT10] WOJTAN C., THÜREY N., GROSS M., TURK G.: Physics-inspired topology changes for thin fluid features. In *SIGGRAPH '10: ACM SIGGRAPH 2010 papers* (New York, NY, USA, 2010), ACM, pp. 1–8.

[ZZQW11] ZHANG G., ZHU D., QIU X., WANG Z.: Skeleton-based control of fluid animation. *The Visual Computer 27*, 3 (2011), 199–210.